International Journal of Computer Science and Applications,
Vol. 11, No. 2, pp. 19 – 37, 2014

A NOVEL FIREFLY ALGORITHM BASED ANT COLONY OPTIMIZATION FOR SOLVING COMBINATORIAL OPTIMIZATION PROBLEMS

ABDESSLEM LAYEB

MISC lab., University of Constantine , Department of computer science and its applications, Constantine, 25000, Algeria
layeb.univ@gmail.com

ZEYNEB BENAYAD

MISC lab., University of Constantine , Department of computer science and its applications, Constantine, 25000, Algeria
zinebbenayad@gmail.com

Inspired by ant colony optimization algorithm, a new firefly optimization algorithm is presented which we call *firefly colony optimization algorithm (FCO)*. Unlike the standard firefly algorithm, the proposed approach is a distributed, and constructive greedy metaheuristic which uses the positive feedback to construct greedily good solutions, and to avoid a premature convergence to low quality solutions. We have assessed the performance of the proposed methodology on the bin packing problem (BPP). The results found show clearly the feasibility and the effectiveness of the proposed algorithm.

Keywords: Firefly Algorithm; Firefly Colony Optimization; Greedy Algorithm; Bin packing problem.

1. Introduction

In order to solve complex problems, ideas gleaned from natural mechanisms have been exploited to develop metaheuristics. Nature inspired optimization algorithms have been extensively investigated during the last decade paving the way for new computing paradigms. The ultimate goal is to develop systems that have ability to learn incrementally, to be adaptable to their environment, and to be tolerant to noise [Jourdan *et al.* (2009)]. Artificial neural networks, genetic algorithms and swarm intelligence are examples of bio-inspired systems used to this end [Fogel (2001)]. In recent years, optimizing by swarm intelligence has become a research interest to many research scientists of evolutionary computation fields. There are many algorithms based swarm intelligence like Ant Colony Optimization (ACO) [Dorigo and Gambardella (1997)], [Hu *et al.* (2008)], Particle Swarm Optimization (PSO) [Clerc and Kennedy (2002)], [Garcia-Gonzalo and Fernandez-Martinez (2012)], [Parpinelli and Lopes (2011)], Artificial Bee Colony algorithm (ABC) [Karaboga and Basturk (2007)], Artificial Immune Systems

(AIS) [Yang *et al.* (2008)], Eco-Systems Optimization [Nebti and Meshoul (2009)], Gravitational Search Algorithm (GSA) [Sahu *et al.* (2013)], Cuckoo Search Algorithm (CSA), [Yang and Deb (2010a)], [Layeb (2011)], etc.

One of the most recent population based optimization algorithms is the Firefly Algorithm (FA) [Yang (2010a)], [Yang (2009)]. FA is a bio-inspired metaheuristic based on the flashing behavior and the phenomenon of bioluminescent communication of tropical fireflies. There are about two thousand firefly species, and most fireflies produce short and rhythmic flashes. The pattern of flashes is often unique for a particular species. However, the flashing light intensity decreases as the distance increases, this phenomenon inspired Yang to develop the basic FA. Since its appearance, the firefly algorithm has shown its promising effectiveness for various optimization problems [Fister *et al.* (2013)]. Indeed, FA is very competitive to deal with multimodal optimization problems [Yang (2009)], however its performance to deal with discrete optimization problems is not good enough compared to other popular swarm optimization algorithms like PSO algorithm, Ant colony, Artificial Bee Colony, etc. Moreover, its algorithmic complexity is somehow higher than similar algorithms like cuckoo search [Yang and Deb (2010a)], bat algorithms [Yang (2010b)], etc.

The purpose of this paper is to present a novel variant of firefly algorithm based on a new behavior. In this variant, the fireflies are seen as cooperative learning agents like ants in Ant Colony Optimization algorithm [Levine and Ducatelle (2004)]. The proposed algorithm has a good combination of distributed computation, autocatalysis, and constructive greediness to find an optimal solution for combinatorial optimization problems. Like in ant colony, the search for a good solution is achieved by an indirect communication known as stigmergy amongst the fireflies. We have supposed in our algorithm that fireflies emit a phosphorescent substance on their ways which make them brightened. When they arrive at a decision point, they make a probabilistic choice, biased by the intensity of the phosphorescent substance's light they see. This behavior has an autocatalytic effect because in fact that when a firefly chooses a path, it will increase the probability that the corresponding path will be chosen again by other fireflies in the future visits. We assume that, when they return back, the probability of choosing the same path is higher (due to the increase of its brightness). Indeed, more phosphorescent substance will be put on the chosen path, which makes it more attractive for future fireflies. The final graph solution in the firefly colony algorithm looks like roads in a city, more lamps on a road more the road is brightened and greater the intensity of a lamp more the road is brightened too. The proposed algorithm has several characteristic features. The speed is significantly increased because the FCO doesn't make pairwise comparisons between all the fireflies like in the standard firefly algorithm of yang [Yang (2010d)]. Secondly, the performance of FCO is improved by decreasing the randomization capacity of the basic FA so that it decreases gradually as the optima are approaching. Thirdly, the hybridization between the greediness ability of the constructive algorithms, the exploitation and the exploration of the firefly algorithm has led to build a powerful algorithm to deal with discrete optimization problems. Finally, FCO was successfully tested on the bin packing problem (BPP) which is a hard combinatorial problem. Comparing with ant colony, cuckoo search, and other heuristics, the performance of our proposed method is significantly better, and it gives best solutions in low computational time.

This paper is organized as follows. The first part contains the description of the firefly algorithm as it is currently implemented. Subsequent section is devoted to the detailed description of the proposed Firefly Colony Optimization (FCO). In the next part, we introduce briefly the bin packing problem; the application of the FCO for BPP is also presented in this section. This will be followed by the results of our numerical analysis in Section 5. Finally, we conclude the paper in the last Section.

2. Firefly Algorithm State of the art

The Firefly Algorithm was developed by Xin She Yang [Yang and He (2013)], [Yang (2010a)] in late 2007 and 2008 in Cambridge University. Fireflies are simple insects living in groups. The social behavior of fireflies is not just dedicated to foraging but more to reproduction [Fister *et al.* (2013)]. In the core of this behavior, we find the direct communication between the fireflies by means of rhythmic flushing light. However, two fundamental functions of a firefly's flashing are to attract mating partners, and to attract potential prey. In addition, flashing may also serve as a protective warning mechanism. The FA is based on these flashing patterns and behavior of fireflies. For simplicity, we can idealize these flashing characteristics as the following three rules:

(1) Fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
(2) The attractiveness is proportional to the brightness, and they both decrease as their distance increases. Thus for any two flashing fireflies, the less brighter one will move towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly.
(3) The brightness of a firefly is determined by the landscape of the objective function.

In the firefly algorithm, there are two important issues: the variation of light intensity and formulation of the attractiveness. In the simplest case for maximum optimization problems, the brightness I of a firefly at a particular location x can be chosen as. $I(x) \propto f(x)$ However, the attractiveness β is relative; it should be seen in the eyes of the beholder or judged by the other fireflies. In the simplest form, the light intensity *I(x)* varies with the distance r monotonically and exponentially as [Yang (2009)] in the Eq. (1)

$$I(r) = I_0 e^{-\gamma r^2} \tag{1}$$

Sometimes, we may need a function which decreases monotonically at a slower rate. In this case, we can use the approximation giving in [Yang (2009)] by Eq. (2)

$$I(r) = \frac{I_0}{\gamma r^2} \tag{2}$$

As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the variation of attractiveness β with the distance r by Eq. (3)

$$\beta = \beta_0 e^{-\gamma r^2} \tag{3}$$

Where β_0 is the attractiveness at $r = 0$. It is worth pointing out that the exponent γr^2 can be replaced by other functions such as γr^m when $m > 0$.

The distance r between any two fireflies i and j at x_i and x_j, respectively, is the Cartesian distance defined in Eq. (4)

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})} \tag{4}$$

The movement of a firefly i attracted to another more attractive (brighter) firefly j is determined by [Yang and He (2013)] according to Eq.(5)

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{i,j}^2} (x_j^t - x_i^t) + \alpha_t \varepsilon_i^t \tag{5}$$

Where the second term is due to the attraction, the third term is randomization with α_t being the randomization parameter, and ε_i^t is a vector of random numbers drawn from a Gaussian distribution or uniform distribution at time t. If $\beta_0 = 0$, it becomes a simple random walk. On the other hand, if $\gamma = 0$, it reduces to a variant of particle swarm optimization. Furthermore, the randomization ε_i^t can easily be extended to other distributions. The basic Firefly Algorithm is described as follows:

Algorithm1: Original Firefly Algorithm

Objective Function f(x), $x = (x_1, x_2, ..., x_d)^T$
Generate initial population of fireflies x_i $(i = 1, 2, .., m)$
Light intensity I_i *at* x_i *is determined by* $f(x_i)$
Define light absorption coefficient γ
while *t <MaxGeneration* **do**
    for *i = 1 : n* **do**
        for *j = 1 : i* **do**
            if $I_j < I_i$ **then**
                Move firefly i towards firefly j in d-dimension;
            end if
            Attractiveness varies with distance r via $e^{-\gamma r}$
            Evaluate new solutions and update light intensity
        end for
    end for
    Rank the fireflies and find the current best
end while
Post-process results and visualization

The Firefly algorithm is a very efficient metaheuristic. The keys of the efficiency of this algorithm resumed in the three following points [Yang and He (2013)]:

- Automatic subdivision of the whole population into subgroups so that each subgroup can swarm around a local mode. Among all the local modes, there exists the global optimality. Therefore, FA can deal with multimodal optimization naturally.
- FA has the novel attraction mechanism among its multiple agents, and this attraction can speed up the convergence.
- FA can include PSO, DE and SA as its special cases. Therefore, it is no surprise that FA can efficiently deal with a diverse range of optimization problems.

In its basic form, the firefly algorithm was designed to solve continuous problems [Yang (2010c)], [Yang (2010d)], [Gandomi *et al.* (2011)]. However, FA was also adjusted for solving combinatorial optimization problems. Indeed, FA and its variants have been applied for solving different optimization, classification and engineering problems [Fister *et al.* (2013)]. Durkota in his Bachelor thesis has adapted the FA for solving some types of quadratic Assignment Problem (QAP) [Durkota (2011)]. This discrete version gives promising results for the simple QAP instances, while the algorithm often falls into the local optima when solving the hard instance problems. Sayadi et al. presented a discrete version of the FA with local search for makespan minimization in permutation flow shop scheduling problems [Sayadi *et al.* (2010)]. Palit et al. presented a binary firefly algorithm for cryptanalytic attack on the knapsack cryptosystem [Palit et *al.* (2011)], where they proved that binary firefly algorithm is capable of finding correct results more efficiently than GA. Jati et al. developed an evolutionary discrete version of FA (EDFA) for solving symmetric traveling salesman problem [Jati and Suyanto (2011)], the simulation results indicated that the EDFA performed very well for some TSP instances when compared to the memetic algorithm. Hassanzadeh et al., proposed in their work [Hassanzadeh *et al.* (2011)] an image segmentation approach based on maximum variance Intra-Cluster method and Firefly algorithm. Coelho et al. [Coelho *et al.* (2011)] have applied a chaotic firefly algorithm to the reliability-redundancy optimization. Finally and not last, and for image processing, Draa et al. presented an opposition-based firefly algorithm for medical image contrast enhancement [Draa *et al.* (2015)]. In other studies, FA can also be hybridized with other general problem solvers thanks to its characteristics, i.e., multi-modality and faster convergence [Fister *et al.* (2013)]. Indeed, we find in the literature several hybridizations of firefly algorithm with other algorithms, such as Eagle strategy using Lévy walk [Yang and Deb (2010b)], genetic algorithms [Luthra and Pal (2011)], cellular learning automata [Hassanzadeh and Meybodi (2012)], ant colony optimization [Rizk-Allah *et al.* (2013)], local search [Sayadi *et al.* (2010)], and others.

On the other hand, the firefly Algorithm is simple in terms of complexity and it is easy to implement. It has two inner loops when going through the population n, and one outer loop for iteration t [Yang and He (2013)]. Unfortunately, the firefly algorithm performs a full pair-wise comparison, i.e., that is based on the comparison and computation of attractiveness between each firefly in the swarm and all the other fireflies, which can be time consuming for large optimization problems [Baykasoğlu and Ozsoydan (2014)]. Moreover, the firefly algorithm was designed initially to solve the continuous optimization problems. Consequently, its adaptation to discrete optimization problems doesn't always give the desired results especially when we want to solve hard and complicated problems such as the traveling salesman problem and bin packing problem [Jati and Suyanto (2011)], [Durkota (2011)]. Indeed, the discretization of the firefly algorithm causes a loss of information which is unacceptable for certain research domains like medical imagery [Benayad and Djenna (2013)].

To overcome the limitations of the firefly algorithm especially for discrete and combinatorial optimization, we have developed a new cooperative and constructive version of the firefly algorithm called *Firefly Colony Optimization (FCO)*. The proposed algorithm is based on both the behaviors of the ant colony optimization and firefly algorithms. In more detail, the proposed method is presented in the subsequent section.

3. The Proposed Firefly Colony Optimization Algorithm

The Firefly Colony Optimization Algorithm FCO was originally inspired by the ability of the real fireflies to attract each other, and to be attracted by the brightest ones and by sources of light. However, the proposed firefly colony optimization (FCO) is a distributed, and constructive greedy metaheuristic. More formally, given a fully connected graph *(n,E)* with *n* nodes and *E* edges, representing all the potential trajectory solutions, the FCO is able to find the best (optimal) combination of these nodes encoding a feasible solution for a given combinatorial optimization problem. In this metaheuristic, we suppose that the fireflies are able to release phosphorescent substance while flying. The paths can absorb this substance and, in turn, they become luminous. We assume also that, the quantity of the phosphorescent substance emitted by the fireflies, at every iteration, is fixed. Consequently, shorter is the path, higher is its brightness. It is clear, that the shortest path become brighter and more attractive than the others, influencing other fireflies to follow it. After few time, the entire colony will follow this shortest path (fig. 1).

In addition to the above characteristics, fireflies are considered as simple agents that cooperate between themselves to construct good solutions. The proposed FCO is guided by the following assumptions (Algorithm 2):

- The fireflies communicate indirectly using stigmergy by means of the phosphorescent substance.
- They build a solution by moving on the graph's nodes.
- They will have some memory for memorizing their paths.
- They put a phosphorescent substance on each node visited at the end of the complete tour.
- The attractiveness between nodes depends to the distance between them and the quantity of the phosphorescent substance.

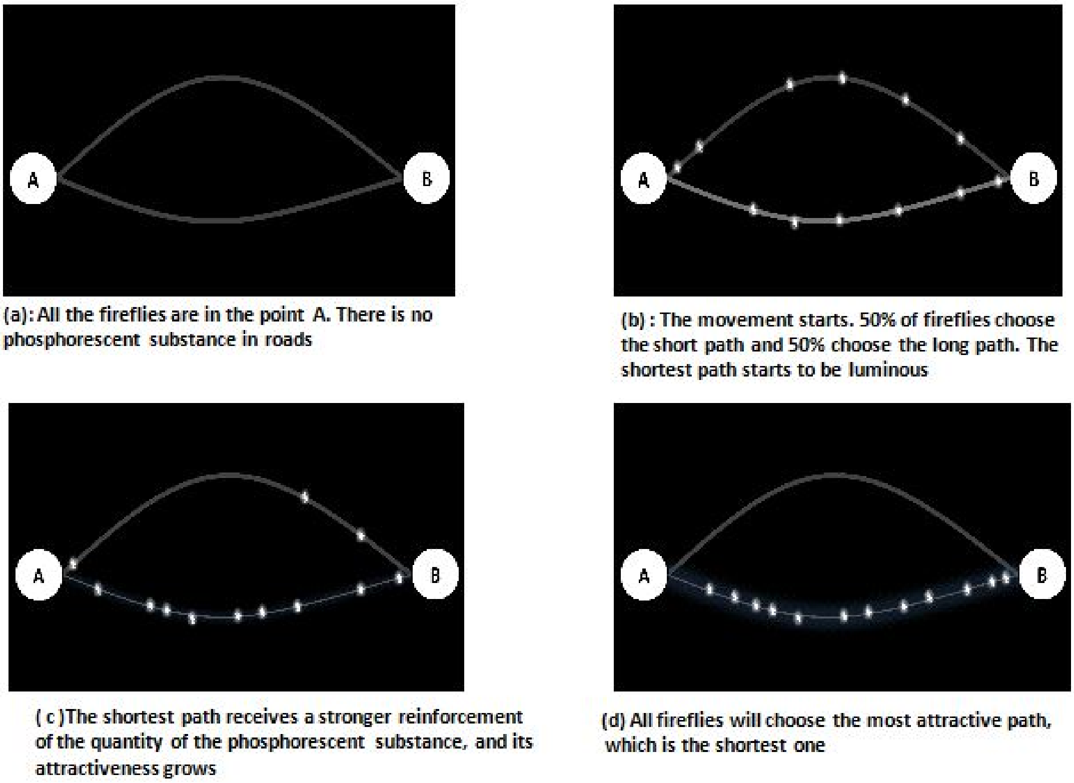

Fig. 1. A theoretical experiment that demonstrates the shortest path finding capability of fireflies.

Algorithm2: Firefly Colony Optimization Algorithm

Objective Function $f(x),\ x = (x_1, x_2, ..., x_d)^T$
Place all fireflies on their start points *{Initialization}*
Initialize the attractiveness matrix
Define light absorption coefficient γ
while t *<MaxGeneration* **do**
  for $k = 1 : n$ **do** *{Solution Construction}*
    s ← constructSolution(k,t) ;
    L←f(s) ;
  end for
  Evaluate new solutions *{evaluation}*
  update the attractiveness matrix and reset the fireflies' memory *{global update}*
end while
Let s^+ *be the shortest path,* L^+ is *the corresponding length*

In more detail, the main steps of the FCO are summarized as follows:

Step one: Initialization

Initially, the fireflies are situated on their start points. The brightness of the graph's nodes is zero owing to the absence of the phosphorescent substance.

Step two: Solution construction

At each iteration, each firefly constructs a solution progressively by choosing the next node to visit using some probability. This later is proportional to the attractiveness, i.e., higher is the attractiveness between the current node and its predecessor node, larger is its probability to be chosen. Like in the standard firefly, the attractiveness between two nodes is inversely proportional to the distance. Consequently, the construction of the solution is formulated by the transition rule (Eq. (6)).

$$\beta_{ij}^{k}\left(t\right)=\begin{cases}\beta_{ij} * e^{-\gamma r_{ij}^{m}}, j \in allowed_{k} \\ 0, \qquad otherwise\end{cases} \tag{6}$$

Where β_{ij}^{k} is the attractiveness at the iteration t between the two nodes i and j, i is the current node where the firefly k is situated, and j is a candidate node which could be chosen as the next node to be inserted in the partial solution built by the firefly k. β_{ij} is the attractiveness between node i and the node j obtained from the attractiveness matrix. γ is the absorption coefficient of the light. r_{ij} is the distance between nodes i and j. $allowed_k$ is the set of the nodes that the firefly k can visit at the iteration t.

In order to choose the next node, two different ways can be used. The first way consists of choosing the node with higher attractiveness. In the second way, we calculate the cumulative sum of the attractiveness of all candidate nodes using the Eq. (7), after that we choose the node having the bigger attractiveness than a random number.

$$vectAttractiveness = cumsum\left(\beta_{ij}^{k}\left(t\right)\right), \quad j \in allowed_{k} \tag{7}$$

Where *cumsum* is the cumulative sum, $\beta_{ij}^{k}(t)$ is the attractiveness calculated using the equation (6). *vectAttractiveness* is an array containing the attractiveness values of the candidates nodes at the iteration t.

Step three: Evaluation

We use an objective function to assess the quality of the complete solutions, and to guide the algorithm towards the good ones. The objective function depends on the problem to be optimized.

Step four: Attractiveness update

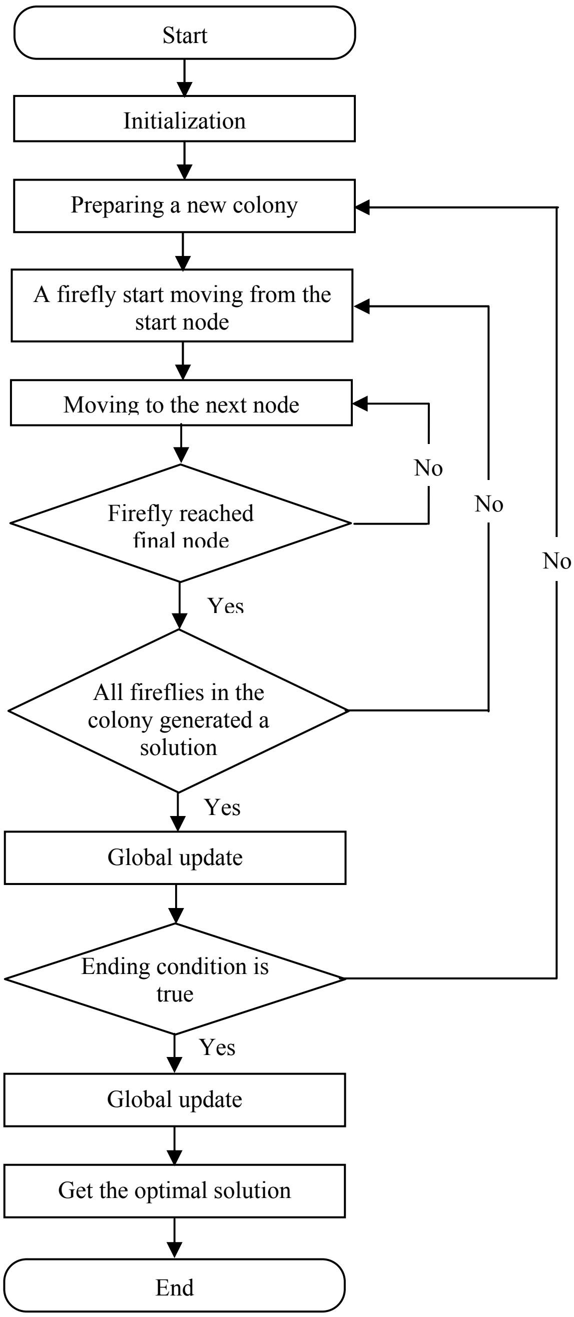

Fig. 2. Flowchart of the Firefly Colony Optimization

After each iteration of the FCO algorithm, each firefly has constructed a solution. Fireflies emit phosphorescent substance while moving on the graph. They put this substance on each node visited at the end of a complete tour. The attractiveness of this substance emitted on each node is calculated by the Eq. (9). It depends on the quality of the solutions of which the node in question is part. At this point the attractiveness is updated according to the Eq. (8). The new attractiveness is the sum of the old one after evaporation of the substance, and the attractiveness of the new phosphorescent substance emitted in the current iteration.

$$\beta_{ij}\left(t+1\right)=\alpha_t \epsilon_i^t * \beta_{ij}\left(t\right)+\sum_{k=1}^{m}\Delta\beta_{ij}^k(t) \tag{8}$$

Where

$$\Delta\beta_{ij}^k\left(t\right)=\begin{cases}\dfrac{Q}{f^k(t)}, & if\ \ (i,j)\epsilon\, solution_k \\ 0 \quad , & otherwise\end{cases} \tag{9}$$

$\beta_{ij}(t)$ is the attractiveness between nodes *i* and *j* at iteration *t*. *Q* is a constant, $f^k(t)$ is the fitness of the solution find by the firefly *k* at the iteration *t*.

The flow-chart of the proposed FCO is given by the figure 2.

4. Firefly Colony Optimization Algorithm for Bin Packing Problem

This section describes how the firefly colony optimization was adapted to solve the Bin Packing Problem (BPP). Section 4.1 presents the BPP, and the section 4.2 is devoted to explain how the FCO was tailored to deal with the bin packing problem.

4.1. *Bin packing problem formulation*

Bin packing problem is an important task in solving many real problems such as the loading of tractor trailer trucks, cargo airplanes and ships, etc. There are three main variants of BPP problems: one, two and three dimensional Bin Packing Problems. In this paper, we deal with the one-dimensional Bin Packing Problem (1-BPP) [Fleszar and Hindi (2002)], [Alvim *et al.* (2004)], [Kao and Lin (1992)]. The 1-BPP consists to pack a set of items having different weights into a minimum number of bins which may have also different capacities. Although, this problem seems to be quite simple to define, it has been shown to be NP-hard, because it cannot be solved accurately and optimally in a reasonable time. These are the reasons why several approximate methods have been proposed to solve this problem, which are generally based on heuristics or metaheuristics. Among the most popular heuristics used to solve the bin packing problem, the First Fit algorithm (FF) which places each item into the first bin in which it will fit. The second popular heuristic algorithm is the Best Fit (BF) which puts each element into the filled bin in which it fits. Moreover, the FF and BF heuristics can be improved by applying a specific order of items like in First Fit Decreasing (FFD) and Best Fit Decreasing (BFD), etc [Monaci (2003)], [Scholl et al. (1997)]. Moreover, many kinds of metaheuristics have been used to solve the bin packing problems like genetic algorithms [Falkenauer (1996)], Ant colony [Wang *et al.* (2010)], etc. Formally, the bin packing problem can be stated as follows:

$$Min\ z(y) = \sum_{j=1}^{n} y_j \tag{10}$$

<u>*Subject to constraints:*</u>

$$\sum_{i=1}^{n} w_i x_{ij} \le c y_i ,\ j = 1..n \tag{11}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1..n \tag{12}$$

$$y_i , x_{ij} \in \{0,1\}, j = 1..n, i = 1..n \tag{13}$$

<u>*With:*</u>

$y_i = 1$ if the bin *i* is used; else 0

$x_{ij} = 1$ if the item *i* is stocked in bin *j;* else 0.

In the above model the objective function is to minimize the total number of bins used to pack all items which have the same capacity (Eq.(10)). The first constraint guarantees that the weights of items (w_i) filled in the bin *j* do not exceed the bin capacity. The second constraint ensures that each item is placed only in one bin. It appears to be impossible to obtain exact solutions in polynomial time. The main reason is that the required computation grows exponentially with the size of the problem. Therefore, it is often desirable to find near optimal solutions to this kind of problems. Efficient heuristic algorithms offer a good alternative to accomplish this goal. Within this perspective, we are interested in applying FCO algorithm to solve this problem.

4.2. *Application of firefly colony optimization algorithm for bin packing problem*

4.2.1. *Construction of a solution*

Every firefly starts with the set of all items to be placed and an empty bin. It will add the items one by one to the opened bin, until none of the items can be added to the current bin. Then, the bin is closed and a new one is opened. The probability that a firefly *k* will choose an item j as the next item to be packed in the current bin *b* of the partial solution *s*, is totally dependent to the attractiveness between the item j and the other items that are already filled in the bin b. The probability is given by Eq. (7).
The attractiveness is formulated by the transition rule (14)

$$\beta(s,b,j) = \begin{cases} \beta_b(j) * e^{\gamma w_j} , & j \in J_k(s,b) \\ 0 , & otherwise \end{cases} \tag{14}$$

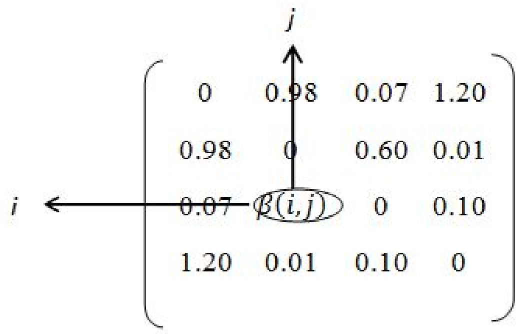

Fig. 3. Flowchart of the Firefly Colony Optimization.

Where:
$\beta(s,b,j)$ is the attractiveness between item j and the items filled in the current bin b in the partial solution s. w_j is the weight of the item j. γ is the absorption coefficient of the light. $J_k(s,b)$ is the set of all items able for inclusion in the current bin.
The brightness $\beta_b(j)$ of an item j in a bin b is given in Eq. (15). It is the sum of all the attractiveness values between item size j and the item sizes i that are already packed in the bin b, divided by the number of items in b for normalization. If b is empty, the brightness $\beta_b(j)$ is set to 1.

$$\beta_j = \begin{cases} \dfrac{\sum_{i \in b} \beta(i,j)}{|b|}, & if \quad b \neq \emptyset \\ 1, & otherwise \end{cases} \qquad (15)$$

In this equation, $\beta(i,j)$ is the attractiveness between the items i and j. This attractiveness is represented by a 2D symmetric matrix as shown in Figure 3.

4.2.2. *Attractiveness updating*

The attractiveness between each pair of items is updated by using the equations 8 and 9.

4.2.3. *Objective function*

The objective function is an important parameter in any optimization algorithm. Indeed, a good objective function helps the search process to find the optimal solution. In the context of the bin packing problem, using the number of bins as objective function can make the algorithm suffer from stagnation because there can be several arrangements which have the same number of bins. Though, this information will be better if it is integrated with other information like the fullness of the bins. That's why we have used the objective function defined by Falkenauer and Delchambre in [Falkenauer and Delchambre (1992)], which contains both item's weight and bin capacities information. This objective function is given by the Eq. (16)

$$Max\ f = (sum_i / C)^k / nbin \tag{16}$$

where: sum_i is the sum of all weight items packed in the bin *i*.
C: is the capacity of bin.
nbin is the number of used bins.
k is the parameter that defines equilibrium of the filling bins. By increasing *k*, we give a higher fitness to solutions that contain a mix of well-filled and less well-filled bins, rather than equally filled bins.

5. Implementation and Validation

This section describes the results of our experiments. First the different parameters values are defined, and then the results of applying FCO for BPP (FCOBPP) are mentioned and discussed. We have implemented our approach in Matlab 7.7 and tested on home PC with core duo processor and 3 GB of memory. In order to evaluate the proposed approach, we have used a set of benchmark data sets taken from the site http://www.wiwi.uni jena.de/Entscheidung/bin-pp/. The benchmark data sets are divided into three classes: easy, medium and hard class instances.
The parameters used in our experimental design are as follows. The number of fireflies *n* is set to 10 and the number of iterations *it* is chosen to be 5; empirically, we have found that 5 iterations are quite enough for obtaining good results. In the update function, the constant *alpha* controlling the randomness is set to 0.2 and the parameter *Q* is set empirically to 2. The parameter *k* in the fitness function is also set to 2 [Levine and Ducatelle (2004)]. The good γ values for the different problems were all situated between zero and one, and in practice, for the medium class instances γ is set to 0.01, for the hard class instances γ is set to 0.001, and 1 for the easy class instances.
The obtained results are summarized in Table 1., Table 2., and Table 3., the first column is the name of the instance, the second column contains the number of items, the third contains the bin capacity, the fourth column contains the results of the first fit decreasing heuristic (FFD), the fifth column contains the results of the quantum cuckoo search algorithm (QICS) [Layeb and Boussalia (2012)], the sixth column contains the results of our approach (FCO), the seventh column contains the results of firefly algorithm with first fit heuristic (FA), the eighth column contains the results of Ant System algorithm (AS), and the last column contains the best known results. Moreover, we have used the Friedman test for comparing statistically the obtained solutions. We should mention that we didn't add the local search to our algorithm.
As it is well seen from the results presented in Figure 4 of Friedman test for the three types of instances (Tables 1, 2 and 3), our FCO algorithm succeeds in finding the near optimal solutions compared to the other methods. Indeed, the FCO results are the closest to the best known solutions unlike to the results found by the other heuristics which are some far from the best known results. We note also that the basic FA is not successful in this experiment. In addition to the solutions quality, our FCO is fast, it founds the optimal solution in less time when compared to the basic FA and QICS, thanks to

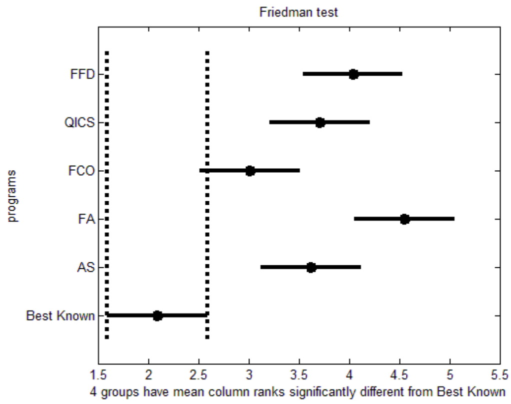

Fig. 4. Friedman test for All -type instances.

characteristics of FCO, there is no pairwise comparison between fireflies like in FA or local search phase like in QICS. Despite that FCO does not use any local search procedure to enhance the results, our proposed algorithm is able to find promising results compared to other metaheuristics.

In more detail, table 1 shows the results of the first series of easy-type instances, the results obtained of FCO, FFD, QICS and AS are completely identical to the best known solutions. However, the results of the basic FA are not good enough (fig.5); FA needs more iteration steps to converge.

Table 1. The bin packing results for the easy class, N: number of items, C: bin capacity

Instance	***N***	***C***	***FFD***	***QICSABP***	***FCO***	***FA***	***AS***	***Best Known***
N1C1W1_A	50	100	25	25	25	26	25	25
N1C1W1_D	50	100	28	28	28	28	28	28
N1C1W1_G	50	100	25	25	25	26	25	25
N1C1W1_B	50	100	31	31	31	31	31	31
N1C1W1_E	50	100	26	26	26	27	26	26
N1C1W1_F	50	100	27	27	27	27	27	27
N1C1W1_I	50	100	25	25	25	25	25	25
N2C1W2_P	100	100	68	68	68	68	68	68
N2C1W2_N	100	100	64	64	64	65	64	64
N2C1W2_O	100	100	64	64	64	66	64	64
N2C1W2_R	100	100	67	67	67	68	67	67
N40C1W2_T	500	100	323	323	323	328	323	323
N4C1W4_C	500	100	365	365	365	368	365	365
N4C1W4_A	500	100	368	368	368	373	368	368
N4C1W4_D	500	100	359	359	359	364	359	359
N4C1W4_B	500	100	349	349	349	356	349	349

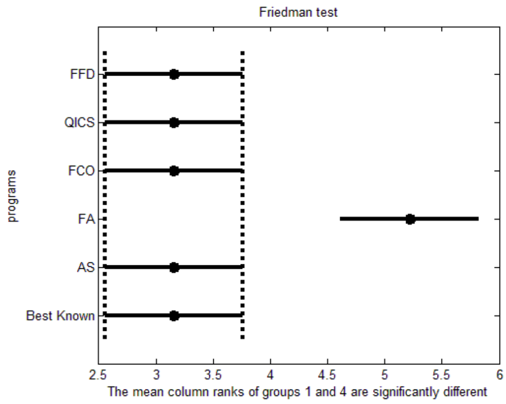

Fig. 5. Friedman test for Easy -type instances.

In the second series of medium instances, our algorithm is better than FA in only one instance. Unlike the previous class, the performance of the basic FA in this class of tests is better, thanks to the performance of FA to deal with multimodal optimization. Indeed, FA is successful in this test (Fig. 6). In other side, FCO is better than the QICSABPP in 9 instances, better than FFD in 10 instances, and better than AS in 6 instances as mentioned in the table 2. In the Freidman test, our approach is the closest to the best know solutions compared to the FFD algorithm, QICS and AS.

Table 2. The bin packing results for the medium class, N: number of items, C: bin capacity

Instance	***N***	***Cap***	***FFD***	***QICSABP***	***FCO***	***FA***	***AS***	***Best Known***
N1W1B2R1	50	1000	18	17	17	17	18	17
N1W1B1R9	50	1000	19	18	17	17	17	17
N1W1B1R2	50	1000	20	20	19	20	20	19
N1W1B2R0	50	1000	18	18	18	18	18	17
N1W1B2R3	50	1000	17	17	17	17	17	16
N2W1B1R0	100	1000	37	36	35	35	37	34
N2W1B1R3	100	1000	38	37	36	36	36	34
N2W1B1R1	100	1000	37	37	36	36	36	34
N2W1B1R4	100	1000	37	37	35	35	35	34
N2W3B3R7	100	1000	13	13	13	13	13	13
N2W4B1R0	100	1000	12	12	12	12	12	12
N4W2B1R0	500	1000	109	109	106	106	107	101
N4W2B1R3	500	1000	109	108	105	105	106	100
N4W3B3R7	500	1000	74	74	74	74	74	74
N4W4B1R0	500	1000	58	58	57	57	58	56
N4W4B1R1	500	1000	58	58	58	58	58	56

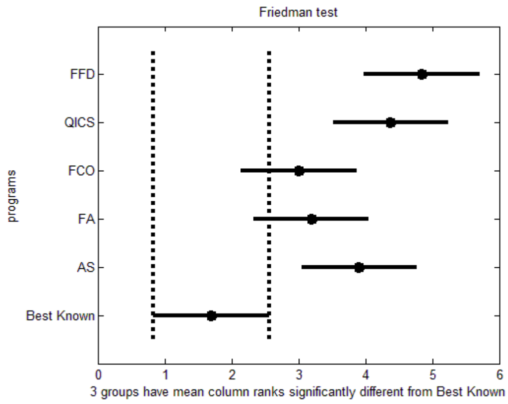

Fig. 6. Friedman test for Medium -type instances.

Table 3 shows the results found for the third series of hard instances. Our algorithm ranks also second in the Friedman test after the best known solutions (fig. 7). In this class, our algorithm is better than the FFD in six instances, better than QICS in three instances, and better than the AS in five instances. The basic FA is not successful in this class; it ranks last in the Friedman test. In this class of tests FA needs a great number of iteration to find good solutions which make it impracticable.

Table 3. The bin packing results for the hard class, N: number of items, C: bin capacity

Instance	*N*	*Cap*	*FFD*	*QICSABP*	*FCO*	*FA*	*AS*	*Best Known*
HARD0	200	100000	59	59	59	60	59	56
HARD1	200	100000	60	60	59	60	60	57
HARD2	200	100000	60	60	59	61	60	56
HARD3	200	100000	59	59	59	60	59	55
HARD4	200	100000	60	60	60	61	60	57
HARD5	200	100000	59	59	59	60	59	56
HARD6	200	100000	60	59	59	61	60	57
HARD7	200	100000	59	59	58	59	59	55
HARD8	200	100000	60	59	59	61	60	57
HARD9	200	100000	60	59	59	60	59	56

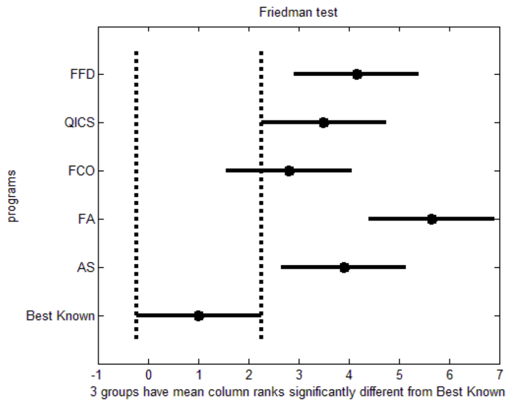

Fig. 7. Friedman test for Hard -type instances.

6. Conclusion

In this paper, we have proposed a new constructive and distributed version of the firefly algorithm which we have called the Firefly Colony Optimization (FCO). FCO combines between the phenomenon of bioluminescent communication of fireflies and foraging behavior of ants. The FCO is very efficient for solving the combinatorial and discrete optimization problems. In order to validate our new algorithm, we have applied it for solving the one-dimensional bin packing problem (BPP). The results are very encouraging and clearly show the effectiveness of our approach. FCO obtains near optimal results with significant faster convergence ability. Therefore, we can say that FCO was found to be more effective compared to standards FA for the discrete optimization problem studied in this paper. As perspective, we want to test the effectiveness of the use of local search methods and difference update brightness rules.

References

Alvim, A. C., Ribeiro, C. C., Glover, F., & Aloise, D. J. (2004). A hybrid improvement heuristic for the one-dimensional bin packing problem. *Journal of Heuristics*, *10*(2), 205-229.

Baykasoğlu, A., & Ozsoydan, F. B. (2014). An improved firefly algorithm for solving dynamic multidimensional knapsack problems. *Expert Systems with Applications*, *41*(8), 3712-3725.

Benayad, Z., Djenna, F.(2013). Variants of the firefly Algorithm for grayscale and color medical image enhancement. Master's thesis, Constantine 2 University.

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, *6*(1), 58-73.

Coelho, L., de Andrade Bernert, D. L., & Mariani, V. C. (2011, June). A chaotic firefly algorithm applied to reliability-redundancy optimization. In *Evolutionary Computation (CEC), 2011 IEEE Congress on* (pp. 517-521). IEEE.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, *1*(1), 53-66.

Draa, A., Benayad, Z. , & Djenna, F. (2015), An Opposition-based Firefly Algorithm for Medical Image Contrast Enhancement, to appear in the International Journal of Information and Communication Technology, Inderscience.

Durkota, K. (2011). Implementation of a discrete firefly algorithm for the QAP problem within the sage framework. *Bachelor Thesis, Czech Technical University*.

Falkenauer, E., Delchambre, A. (May 1992). A genetic algorithm for bin packing and line balancing. In *Proceedings of the IEEE 1992 International Conference on Robotics and Automation*, Nice, France.

Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing.*Journal of heuristics*, *2*(1), 5-30.

Fister, I., JrFister, I., Yang, X. S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*.

Fleszar, K., & Hindi, K. S. (2002). New heuristics for one-dimensional bin-packing. *Computers & operations research*, *29*(7), 821-839.

Fogel, D. B. (2001). An introduction to evolutionary computation, tutorial. In*Congress on Evolutionary Computation (CEC'2001), Seoul, Korea*.

Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, *89*(23), 2325-2336.

Garcia-Gonzalo, E., & Fernandez-Martinez, J. L. (2012). A brief historical review of particle swarm optimization (PSO). *Journal of Bioinformatics and Intelligent Control*, *1*(1), 3-16.

Hassanzadeh, T., Vojodi, H., & Moghadam, A. M. E. (2011, July). An image segmentation approach based on maximum variance Intra-cluster method and Firefly algorithm. In *Natural Computation (ICNC), 2011 Seventh International Conference on* (Vol. 3, pp. 1817-1821). IEEE.

Hassanzadeh, T., & Meybodi, M. R. (2012). A new hybrid algorithm based on Firefly Algorithm and cellular learning automata. In *20th Iranian Conference on Electrical Engineering (ICEE)* (pp. 628-633).

Hu, X. M., Zhang, J., & Li, Y. (2008). Orthogonal methods based ant colony search for solving continuous optimization problems. *Journal of Computer Science and Technology*, *23*(1), 2-18.

Jati, G. K., & Suyanto, S. (2011). Evolutionary discrete firefly algorithm for travelling salesman problem. In *Adaptive and Intelligent Systems* (pp. 393-403). Springer Berlin Heidelberg.

Jourdan, L., Basseur, M., & Talbi, E. G. (2009). Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*,*199*(3), 620-629.

Kao, C. Y., & Lin, F. T. (1992, October). A stochastic approach for the one-dimensional bin-packing problems. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on* (pp. 1545-1551). IEEE.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, *39*(3), 459-471.

Layeb, A. (2011). A novel quantum inspired cuckoo search for knapsack problems. *International Journal of Bio-Inspired Computation*, *3*(5), 297-305.

Layeb, A., & Boussalia, S. R. (2012). A Novel Quantum Inspired Cuckoo Search Algorithm for Bin Packing Problem. *International Journal of Information Technology and Computer Science (IJITCS)*, *4*(5), 58.

Levine, J., & Ducatelle, F. (2004). Ant colony optimization and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society*, *55*(7), 705-716.

Luthra, J., & Pal, S. K. (2011, December). A hybrid firefly algorithm using genetic operators for the cryptanalysis of a monoalphabetic substitution cipher. In *Information and Communication Technologies (WICT), 2011 World Congress on* (pp. 202-206). IEEE.

Monaci, M. (2003). Algorithms for packing and scheduling problems. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, *1*(1), 85-87.

Nebti, S., & Meshoul, S. (2009). Predator prey optimization for snake-based contour detection. *International Journal of Intelligent Computing and Cybernetics*, *2*(2), 228-242.

Palit, S., Sinha, S. N., Molla, M. A., Khanra, A., & Kule, M. (2011, September). A cryptanalytic attack on the knapsack cryptosystem using binary Firefly algorithm. In *Computer and Communication Technology (ICCCT), 2011 2nd International Conference on* (pp. 428-432). IEEE.

Parpinelli, R. S., & Lopes, H. S. (2011). New inspirations in swarm intelligence: a survey. *International Journal of Bio-Inspired Computation*, *3*(1), 1-16.

Rizk-Allah, R. M., Zaki, E. M., & El-Sawy, A. A. (2013). Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems.*Applied Mathematics and Computation*, *224*, 473-483.

Sahu, R. K., Panda, S., Rout, U. K., & Raul, P. (2013). Application of Gravitational Search Algorithm for Load Frequency Control of Multi Area Power System. *Journal of Bioinformatics and Intelligent Control*, *2*(3), 200-210.

Sayadi, M., Ramezanian, R., & Ghaffari-Nasab, N. (2010). A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, *1*(1).

Scholl, A., Klein, R., & Jürgens, C. (1997). BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers & Operations Research*, *24*(7), 627-645.

Wang, S., Wang, L., Shi, R., & Ge, M. (2010, June). Study on improved ant colony optimization for bin-packing problem. In *Computer Design and Applications (ICCDA), 2010 International Conference on* (Vol. 4, pp. V4-489). IEEE.

Yang, H., Elhadef, M., Nayak, A., & Yang, X. (2008, December). Network fault diagnosis: an artificial immune system approach. In *Parallel and Distributed Systems, 2008. ICPADS'08. 14th IEEE International Conference on* (pp. 463-469). IEEE.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *Stochastic algorithms: foundations and applications* (pp. 169-178). Springer Berlin Heidelberg.

Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver Press.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Springer Berlin Heidelberg.

Yang, X. S. (2010). Firefly algorithm, Levy flights and global optimization. In*Research and Development in Intelligent Systems XXVI* (pp. 209-218). Springer London.

Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, *2*(2), 78-84.

Yang, X. S., & Deb, S. (2010). Engineering optimisation by cuckoo search.*International Journal of Mathematical Modelling and Numerical Optimisation*,*1*(4), 330-343.

Yang, X. S., & Deb, S. (2010). Eagle strategy using Levy walk and firefly algorithms for stochastic optimization. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (pp. 101-111). Springer Berlin Heidelberg.

Yang, X. S., & He, X. (2013). Firefly algorithm: recent advances and applications. *International Journal of Swarm Intelligence*, *1*(1), 36-50.